Video Compression/Decompression Pipeline

SDMay24-12

Team Members: Benjamin Meinders, Caleb Rock, Colsen Selk, Kareem Eljaam, and Logan McDermott
Faculty Advisor: Dr. Joseph Zambreno
Client: John Deere

IOWA STATE UNIVERSITY

Problem Statement

This project's goal is to increase the amount of computer vision workload that can be handled by an FPGA while reducing the on-chip RAM usage by using pipelineable compression and decompression cores.

Users

John Deere will use this project for video processing and computer vision applications.

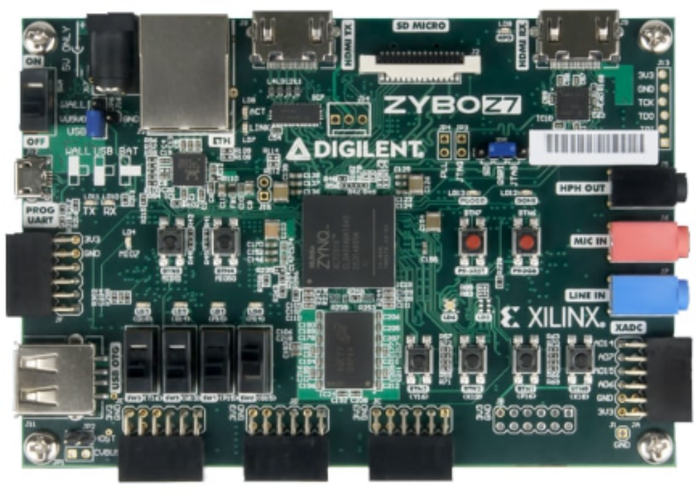
Standards

- Python for prototyping
- Java and C++ for more involved testing
- VHDL for FPGA IPs
- Vitis HLS
- LZW video compression and BWT data transformation
- AXI-Stream Video Formatting
- HDMI for transmission of data
- Standardized data-size formats (the Byte)

Hardware Requirements

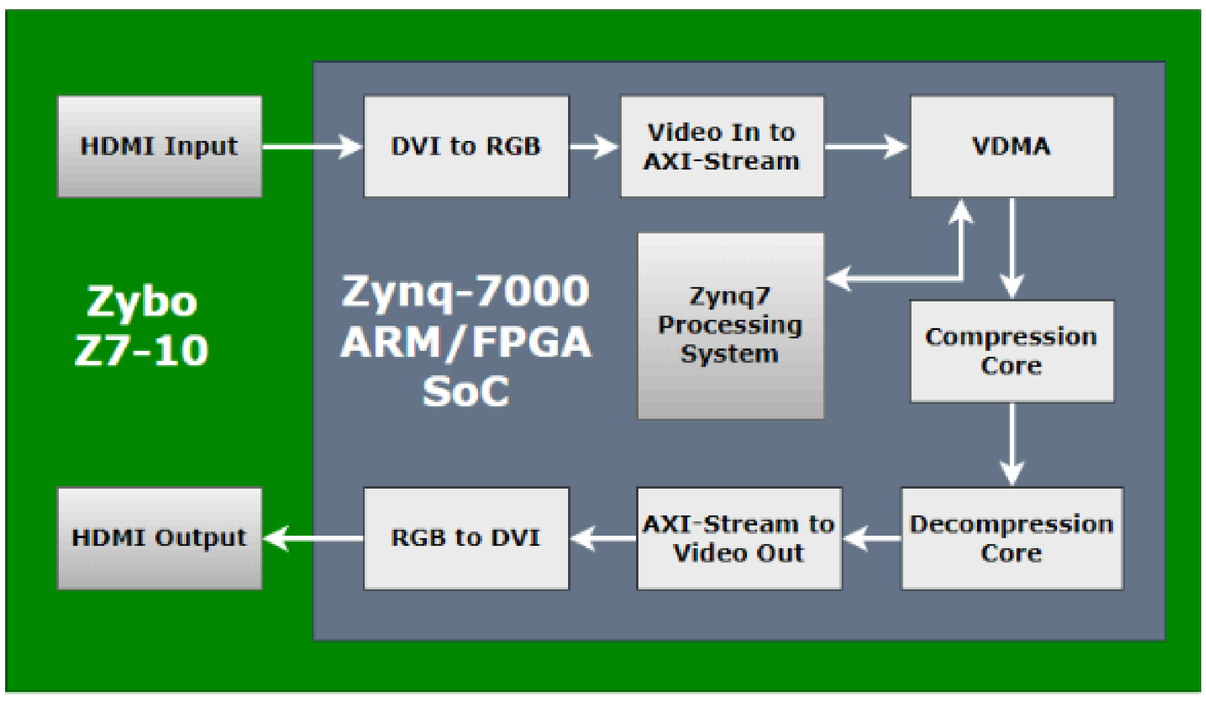
- Create lightweight compression and decompression cores
- Compression and decompression should have as little loss of data as possible
- Compression and decompression are to be at a resolution of at least 1920x1080
- Prioritize lower latency over compressibility
- Solution must be pipelineable

Zybo Z7-10



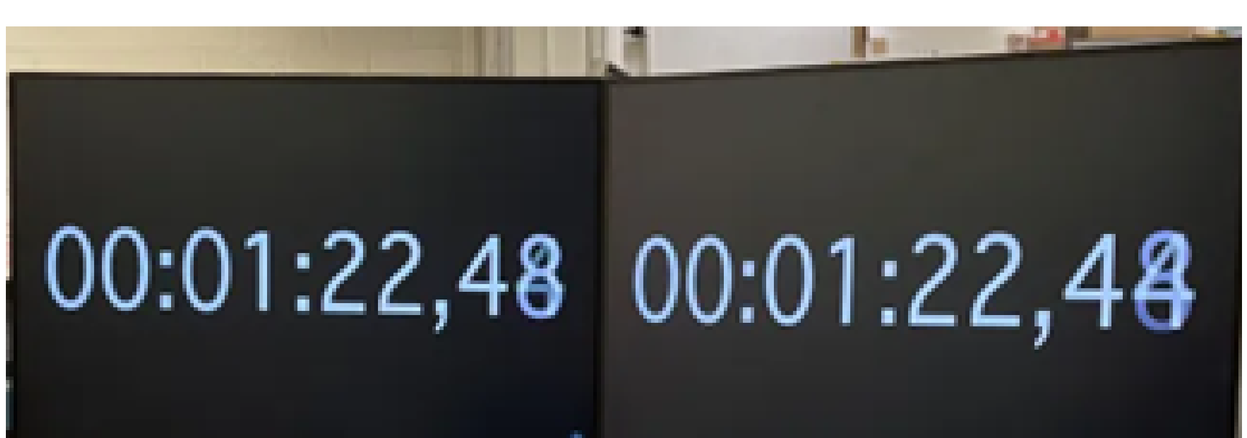
The Z7-10 includes a Zynq-7000 FPGA and 2 HDMI ports which make it perfect for testing passthrough through HDMI

Design



Testing

- LZW & BWT tested with java and python
- LZW & BWT found to not be sufficient compression, but were shown to be theoretically fast enough if pipelined
- For RGB values, a bit size of 5-5-5 was initially tried, then 6-6-4, then 5-6-5.
- 5-6-5 gave best results in terms of image quality



Technical Details

- Throughput = 402 MB/s
- Compression Ratio = 1.5
- RGB24 Pixel Size = 24
- RGB16 Compressed Pixel Size = 16
- RAM requirement is now only 2/3 of the original RAM reqiurement whle video is mostly maintained

Compression Scheme